

## REMARKS/ARGUMENTS

### Status of the Claims

- Claims 1-87 are pending in the Application after entry of this amendment.
- Claims 1-87 are rejected by Examiner.
- Claims 1, 17, 24, 40, 56, and 72 are amended by Applicants.

### Claim Rejections Pursuant to 35 U.S.C. §102(e)

Claims 1-7, 12-14, 16-30, 35-37, 39-46, 51-53, 55-62, 67-69, 71-78, 83-85 and 87 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Brumme (U.S. Patent No. 6,134,559).

Claims 1, 17, 24, 40, 56, and 72 as amended include features that are neither disclosed nor suggested by the prior art, namely, as represented by claim 1:

A method for operating a computer using object-based computer code, the method comprising:

invoking an event handler method, using a delegate, by calling another method of an instance of a class for which parameters passed to the other method are also passed to the event handler method, a parameter list of the other method having a same signature as a parameter list of the event handler method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;

creating an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; and  
dynamically altering contents of the invocation list. (emphasis added)

Claims 1, 17, 24, 40, 56, and 72 were amended to clarify that an event handler method is invoked using a delegate by calling another method, wherein the other method references the event handler, and the delegate contains a reference to the other method. The present invention uses a delegate-based event driven programming model to handle events (Application as filed, page 10, ll. 3-4). A delegate is an object that contains a pointer to a method, as well as a pointer to an object that the referenced method is to be applied (*Id.* at page 10, ll. 5-7).

Brumme neither describes nor suggests invoking an event handler method using a delegate by calling another method, wherein the other method references the event handler, and the delegate contains a reference to the other method. Brumme describes a system and

method for integrating objects of foreign type systems into a single unified type system (Brumme at col. 4, ll. 2-4). Foreign object adapters are configured to support one or more foreign object systems (*Id.* at col. 4, ll. 19-21). The foreign object adapters convert a class type from the foreign object system to a class type compatible with the single unified type system (*Id.* at col. 4, ll. 39-41). It is respectfully submitted that claims 1, 17, 24, 40, 56, and 72 as amended are now in condition for allowance. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 102(e) rejection and allow claims 1, 17, 24, 40, 56, and 72.

With respect to dependant claims 2-7, 12-14, 16, 18-30, 35-37, 39, 41-46, 51-53, 55, 57-62, 67-69, 71, 73-78, 83-85, and 87, Applicant submits that they are now allowable for the reasons set forth above. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 102(e) rejection and allow the claims.

**Claim Rejections Pursuant to 35 U.S.C. §103 (a)**

Claims 8-11, 31-34, 47-49, 50, 63-66 and 79-82 stand rejected pursuant to 35 U.S.C. § 103(a) as being unpatentable over Brumme in view of Wold (U.S. Patent No. 5,724,589). Applicant respectfully submits that these claims are variously dependant on the independent claims and are now allowable for the reasons set forth above. Wold fails to cure the deficiencies of Brumme. Wold merely describes a development system for creating context-free reusable software components (Wold at col. 3, ll. 59-62). Wold fails to describe invoking an event handler method using a delegate by calling another method, wherein the other method references the event handler, and the delegate contains a reference to the other method as recited by the claims. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 103(a) rejection and allow the claims.

Claims 15, 38, 54, 70 and 86 stand rejected pursuant to 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,134,559 to Brumme in view of Kimura (U.S. Patent No. 6,292,849). Applicant respectfully submits that these claims are variously dependant on the independent claims and are now allowable for the reasons set forth above. Kimura fails to cure the deficiencies of Brumme. Kimura merely describes a system for sharing objects by multiple application programs (Kimura at col. 2, ll. 13-15). The system comprises plural

**DOCKET NO.:** MSFT-0568/160310.1  
**Application No.:** 09/808,263  
**Office Action Dated:** December 11, 2003

**PATENT**

control objects used as common objects, each object further comprising a first function for serving properties to an application program, another control object, and a second function for posting events (*Id.* at col. 2, ll. 15-20). Kimura fails to describe invoking an event handler method using a delegate by calling another method, wherein the other method references the event handler, and the delegate contains a reference to the other method as recited by the claims. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 103(a) rejection and allow the claims.

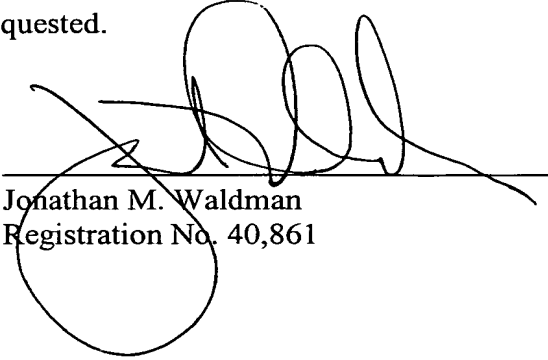
**DOCKET NO.:** MSFT-0568/160310.1  
**Application No.:** 09/808,263  
**Office Action Dated:** December 11, 2003

**PATENT**

**Conclusion**

In view of the above amendments and remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested.

Date: March 24, 2004



Jonathan M. Waldman  
Registration No. 40,861

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439